Malachi Eberly

Project 3 Report

Model Architecture

For my model, I used the LLM's recommendation of a cone-shaped Generator and Discriminator, where the Generator shrunk in layer size and the Discriminator increased again (until the single neuron output layer). The fixed noise had a dimension of 100, which matched the Generator's input layer. I used BCELoss and the Adam optimizer because they seem to have performed well in past assignments. The learning rate of 0.0002 and betas of (0.5, 0.999) that the LLM chose performed well, so I kept them.

Training Process

The training process was very slow but simple to implement. I asked the LLM for an appropriate number of epochs based on the size of the training set, and I decided on 50 for a balance of complete training and the needed training time. Each epoch took a long time because of the convolutional nature of the model and my computer's use of a CPU. Once training was working, I implemented plots to show the images after each epoch and a plot to show the losses after training was finished.

Encountered Challenges

The biggest challenge I encountered was with matching tensor sizes. I ended up asking the LLM to write a custom dataset class. When I did that, it had trouble with adding the batch size to the real_cpu data tensor. It eventually figured that the training loop should use real_cpu = data.to(device) instead of real_cpu = data[0].to(device). Besides that, the code implementation was very straight forward.

LLM Decision and Challenges

I decided on creating a custom GPT using GPT 4 that was tuned for this assignment. It was very efficient with its code implementation, and I could work with it easily to tune things like the number of epochs and the model architecture. I had to have a little bit of back and forth with the LLM to add features such as the device, and sometimes it gave suggestions on what could cause bugs instead of actually looking for bugs, but with a couple of attempts, I was able to have it find and address the small errors in the code.